

RRRRRRRRRRRR		UUU		UUU	NNN	NNN	000000000		FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU		UUU	NNN	NNN	000000000		FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU		UUU	NNN	NNN	000000000		FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRR	RRR	UUU		UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNNNNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNNNNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNNNNN	NNN	000	000	FFF	FFF
RRRRRRRRRRRR		UUU		UUU	NNN	NNN	000	000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU		UUU	NNN	NNN	000	000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU		UUU	NNN	NNN	000	000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRR	RRR	UUU		UUU	NNN	NNNNNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNNNNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNNNNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUUUUUUUUUUUUUUU		UUUUUUUUUUUUUUUU	NNN	NNN	000000000		FFF	FFF
RRR	RRR	UUUUUUUUUUUUUUUU		UUUUUUUUUUUUUUUU	NNN	NNN	000000000		FFF	FFF
RRR	RRR	UUUUUUUUUUUUUUUU		UUUUUUUUUUUUUUUU	NNN	NNN	000000000		FFF	FFF

NN		NN	MM		MM	
NN		NN	MM		MM	
NN		NN	MMMM		MMMM	
NN		NN	MMMM		MMMM	
NNNN		NN	MM	MM	MM	
NNNN		NN	MM	MM	MM	
NN	NN	NN	MM		MM	
NN	NN	NN	MM		MM	
NN		NNNN	MM		MM	
NN		NNNN	MM		MM	
NN		NN	MM		MM	
NN		NN	MM		MM	...
NN		NN	MM		MM	...
NN		NN	MM		MM	...
NN		NN	MM		MM	...

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```



```
0001 0 %TITLE 'Processes the various .NUMBER directives.'
0002 0 MODULE nm ( IDENT = 'V04-000'
P 0003 0 %BLISS32[,ADDRESSING_MODE (EXTERNAL = long_relative,
0004 0 NONEXTERNAL = long_relative)]
0005 0 ) =
0006 1 BEGIN
0007 1
0008 1 *****
0009 1 *
0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0012 1 * ALL RIGHTS RESERVED.
0013 1 *
0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0019 1 * TRANSFERRED.
0020 1 *
0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0023 1 * CORPORATION.
0024 1 *
0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0027 1 *
0028 1 *
0029 1 *****
0030 1
0031 1 ++
0032 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
0033 1
0034 1 ABSTRACT: Processes .NUMBER PAGE, .NUMBER SUBPAGE,
0035 1 .NUMBER INDEX, .NUMBER CHAPTER,
0036 1 .NUMBER APPENDIX, .NUMBER LIST, and
0037 1 .NUMBER LEVEL commands.
0038 1
0039 1 Also, for DSRPLUS:
0040 1 .NUMBER EXAMPLE, .NUMBER FIGURE, and
0041 1 .NUMBER TABLE.
0042 1
0043 1 ENVIRONMENT: Transportable
0044 1
0045 1 AUTHOR: R.W.Friday CREATION DATE: June, 1978
0046 1
```

NM
V04-000

Processes the various .NUMBER directives.
Revision History

K 4
16-Sep-1984 01:16:58
14-Sep-1984 13:07:25

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]NM.BLI;1

Page 2
(2)

:	48	0047	1	%SBTTL 'Revision History'
:	49	0048	1	
:	50	0049	1	MODIFIED BY:
:	51	0050	1	
:	52	0051	1	014 REM00014 Ray Marshall 27-April-1983
:	53	0052	1	Almost complete redesign of the logic that processes the
:	54	0053	1	.NUMBER APPENDIX and .NUMBER CHAPTER directives. This was
:	55	0054	1	done to decouple these directives from the .APPENDIX and
:	56	0055	1	.CHAPTER directives. The result herein was a significant
:	57	0056	1	reduction in code.
:	58	0057	1	
:	59	0058	1	013 KAD00013 Keith Dawson 07-Mar-1983
:	60	0059	1	Global edit of all modules. Updated module names, idents,
:	61	0060	1	copyright dates. Changed require files to BLISS library.
:	62	0061	1	
:	63	0062	1	--


```

65      0063 1 %SBTTL 'Module Level Declarations'
66      0064 1
67      0065 1 | TABLE OF CONTENTS:
68      0066 1 |
69      0067 1 | INCLUDE FILES:
70      0068 1 |
71      0069 1 |
72      0070 1 LIBRARY 'NXPORT:XPORT';      ! XPORT Library
73      0071 1 REQUIRE 'REQ:RNODEF';      ! RUNOFF variant definitions
74      0202 1
75      U 0203 1 %IF DSRPLUS %THEN
76      U 0204 1 LIBRARY 'REQ:DPLLIB';      ! DSRPLUS BLISS Library
77      0205 1 %ELSE
78      0206 1 LIBRARY 'REQ:DSRLIB';      ! DSR BLISS Library
79      0207 1 %FI
80      0208 1
81      0209 1 |
82      0210 1 | EXTERNAL REFERENCES:
83      0211 1 |
84      0212 1 EXTERNAL
85      0213 1
86      0214 1 ECC          : $ECC_BLOCKVECTOR,
87      0215 1 FS01        : FIXED_STRING,
88      U 0216 1 %IF DSRPLUS %THEN
89      U 0217 1 FLGT       : FLGT_DEFINITION,
90      0218 1 %FI
91      0219 1 GCA         : GCA_DEFINITION,
92      0220 1 HCT         : HCT_DEFINITION,
93      0221 1 HLLIST      : COUNTED_LIST,
94      0222 1 IRA         : FIXED_STRING,
95      U 0223 1 %IF DSRPLUS %THEN
96      U 0224 1 khar,
97      0225 1 %FI
98      0226 1 LSTCNT      : REF COUNTED_LIST,
99      0227 1 NMLST       : NUMBER_LIST,
100     0228 1 NPAGEN      : PAGE_DEFINITION,
101     0229 1 NUMPRM      : NUMPRM_DEFINE,
102     0230 1 PAGEN       : PAGE_DEFINITION,
103     0231 1 PHAN        : PHAN_DEFINITION;
104     0232 1
105     0233 1 EXTERNAL LITERAL      ! Error messages
106     0234 1 RNFILC,
107     0235 1 RNFINM;
108     0236 1
109     0237 1 EXTERNAL ROUTINE
110     U 0238 1 %IF DSRPLUS %THEN
111     U 0239 1 GETSUB,
112     0240 1 %FI
113     0241 1 GETLET,
114     0242 1 CONVLB,
115     0243 1 ERMA,
116     0244 1 GLNM,
117     0245 1 GSLU,
118     0246 1 RSKIPS;
119     0247 1
```

```
121 0248 1 %SBTTL 'NM -- main routine'
122 0249 1 GLOBAL ROUTINE nm (HANDLER_CODE) : NOVALUE =
123 0250 1
124 0251 1 ++
125 0252 1 FUNCTIONAL DESCRIPTION:
126 0253 1
127 0254 1     See the ABSTRACT, above.
128 0255 1
129 0256 1 FORMAL PARAMETERS:
130 0257 1
131 0258 1     HANDLER_CODE - Indicates which command is to processed.
132 0259 1
133 0260 1 IMPLICIT INPUTS:
134 0261 1
135 0262 1     NUMPRM - Contains a number, as processed by GETNUM.
136 0263 1
137 0264 1 IMPLICIT OUTPUTS:      None
138 0265 1
139 0266 1 ROUTINE VALUE:
140 0267 1 COMPLETION CODES:      None
141 0268 1
142 0269 1 SIDE EFFECTS:            None
143 0270 1 --
144 0271 1
145 0272 2 BEGIN
146 0273 2
147 0274 2 ! Except for .NUMBER LIST, .NUMBER LEVEL, and .NUMBER RUNNING, all these
148 0275 2 ! directives turn on page numbering on the next page, at the latest.
149 0276 2
150 0277 2 IF (.HANDLER_CODE NEQ H_NUMBER_LIST) AND
151 0278 2     (.HANDLER_CODE NEQ H_NUMBER_LEVEL) AND
152 0279 2     (.HANDLER_CODE NEQ H_NUMBER_RUNNIN)
153 0280 2 THEN
154 0281 2 BEGIN
155 0282 2     HCT_NMPG_NP = TRUE;
156 0283 2
157 0284 2     ! At the top of the first page, or in the middle of a page
158 0285 2     ! turn on page numbering immediately.
159 0286 2
160 0287 2 IF .PHAN_TOP_FIRST OR NOT .PHAN_TOP_PAGE
161 0288 2 THEN
162 0289 2     HCT_NUMBER_PAGE = TRUE;
163 0290 2 END;
164 0291 2
165 0292 2 ! Process the specified command.
166 0293 2 SELECTONE .HANDLER_CODE OF
167 0294 2 SET
168 0295 2
169 0296 2     [H_NUMBER_APPEND, H_NUMBER_CHAPTE] :
170 0297 2     BEGIN
171 0298 2
172 0299 2         LOCAL
173 0300 2             section_number;
174 0301 2
175 0302 2         ! Ignore command if an illegal number was given.
176 0303 2         IF NOT .num_result
177 0304 2         THEN
```



```
178      RETURN;
179
180      section_number = 0;
181
182      ! Try to get a string of letters if the user didn't supply a number.
183      IF .num_length EQL 0
184      THEN
185          BEGIN
186              U      %IF DSRPLUS %THEN
187                  IF (.khar EQL .flgt [sub_flag, flag_character]) AND
188                      .flgt [sub_flag, flag_enabled]
189                  THEN
190                      getsub (ira, num_value, num_length, true)
191                  ELSE
192                      U      %FI
193                      getlet(ira, num_value, num_length);
194              END;
195
196      section_number = .num_value;
197
198      IF (.section_number EQL 0) AND
199          (.num_length EQL 0)
200      THEN
201          ! User said .NUMBER APPENDIX or .NUMBER CHAPTER and he didn't
202          ! specify a number. So he's effectively said nothing new,
203          ! so return.
204          RETURN;
205
206      ! Distinguish between an absolute setting and an adjustment.
207      IF .NUM_SIGN
208      THEN
209          BEGIN
210              ! User gave an adjustment.
211              IF .handler_code EQL h_number_append THEN
212                  section_number = ecc [append_offset, ecc$h_counter] + .section_number
213              ELSE
214                  section_number = ecc [chap_offset, ecc$h_counter] + .section_number;
215              IF .section_number LSS 0 THEN section_number = 0
216          END;
217
218      IF .handler_code EQL h_number_append
219      THEN
220          ecc [append_offset, ecc$h_counter] = .section_number - 1
221      ELSE
222          ecc [chap_offset, ecc$h_counter] = .section_number - 1;
223
224      END;
225
226      [H_NUMBER_INDEX] :
227      BEGIN
228          ! NOTE: It is sufficient to set SCT_TYP as shown to get the
229          !         page numbering to be done correctly. However, SCT_NUMBER
230          !         must be cleared if PAGEQL is not to screw up later.
231
232      IF .PHAN_TOP_FIRST
233      THEN
234          ! At top of first page this takes effect immediately.
```

NM
V04-000

Processes the various .NUMBER directives.
NM -- main routine

B 5
16-Sep-1984 01:16:58
14-Sep-1984 13:07:25

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]NM.BLI;1 Page 6
(4)

**F

```

235      0362 4      BEGIN
236      0363 4      PAGEN [SCT_NUMBER] = 0;
237      0364 4      PAGEN [SCT_TYP]   = SCT_INDEX;
238      0365 4      END;
239      0366 4
240      0367 4      NPAGEN [SCT_NUMBER] = 0;
241      0368 4      NPAGEN [SCT_TYP]   = SCT_INDEX;
242      0369 4      END;
243      0370 4
244      0371 4      [H_NUMBER_LEVEL] :
245      0372 4      BEGIN
246      0373 4      ! Get no more numbers than there are header levels.
247      0374 4      GLNM (.HLLIST [CL_MAX_INDEX]);
248      0375 4
249      0376 4      INCR I FROM 1 TO .NMLST_COUNT DO
250      0377 4      BEGIN
251      0378 4      HLLIST [CL_INDEX] = .I;
252      0379 4
253      0380 4      CASE .NMLST_DESCR (.I) FROM 0 TO 4 OF
254      0381 4      SET
255      0382 4
256      0383 4      [NM_BAD] :
257      0384 4      0;
258      0385 4
259      0386 4      [NM_UNSIGNED] :
260      0387 4      HLLIST [.I] = .NMLST_VALUE (.I);
261      0388 4
262      0389 4      [NM_NULL] :
263      0390 4      0;
264      0391 4
265      0392 4      [NM_PLUS, NM_MINUS] :
266      0393 4
267      0394 4      IF HLLIST [.I] + .NMLST_VALUE (.I) GEQ 0
268      0395 4      THEN
269      0396 4      HLLIST [.I] = .HLLIST [.I] + .NMLST_VALUE (.I)
270      0397 4      ELSE
271      0398 4      ERMA (RNFINM, FALSE);
272      0399 4
273      0400 4      TES;
274      0401 4
275      0402 4      END;
276      0403 4
277      0404 4      HLLIST [.HLLIST [CL_INDEX]] = .HLLIST [.HLLIST [CL_INDEX]] - 1;
278      0405 4      END;
279      0406 4
280      0407 4      [H_NUMBER_LIST] :
281      0408 4      BEGIN
282      0409 4      LOCAL
283      0410 4      LIST_DEPTH;
284      0411 4      ! Set up defaults. It's ok to have no numbers,
285      0412 4      ! so that will never get checked for.
286      0413 4      NMLST_DESCR (1) = NM_NULL;
287      0414 4      NMLST_DESCR (2) = NM_NULL;
288      0415 4      GLNM (2);
289      0416 4      ! Now get parameters
290      0417 4
291      0418 4      ! Sort out the following command formats:
291      0418 4      ! .NMLS
```



```
292 0419 | .NMLS 1,2
293 0420 | .NMLS foobar
294 0421 | .NMLS 2
295 0422 | .NMLS foobar
296 0423 | .NMLS 1,foobar
297 0424 | .NMLS 1 foobar
298 0425 | All the various formats will be put into the "two parameter" format.
299 0426
300 0427 | This code checks to see if the GLNM scan stopped at character string foobar.
301 0428 | The reason for this is that the number list scanning stops when it doesn't find a number.
302 0429 | We want to allow the user to specify a letter string as a counter in this command.
303 0430
304 0431 IF (.NMLST_COUNT EQL 0) OR ! Found absolutely nothing resembling a number. MIGHT be .NMLS fooba
305 0432 | (.NMLST_COUNT EQL 1) OR | Might be .NMLS 1 foobar i.e., missing comma.
306 0433 | ((.NMLST_COUNT EQL 2) AND | There were two arguments given, but
307 0434 | (.NMLST_DESCR (2) EQL NM_NULL)) ! couldn't locate the second number. Might be .NMLS 1, fooba
308 0435 THEN
309 0436 | Check to see if the reason nothing was found is that
310 0437 | a string of letters was given as a counter, instead of just a number.
311 0438 BEGIN
312 0439 FS_INIT (FS01); ! Initialize temporary fixed string.
313 0440
314 0441 | Now, try to get a string of letters
315 0442
316 0443 IF GSLU (IRA, FS01) EQL GSLU_NORMAL
317 0444 THEN
318 0445 | Guessed right! The user gave a string of letters as a counter.
319 0446 | Now convert to their numerical equivalent.
320 0447 BEGIN
321 0448 NMLST_VALUE (2) = CONVLB (.FS_START (FS01), .FS_LENGTH (FS01));
322 0449 | Fake out the following code by telling it two arguments were given.
323 0450 NMLST_DESCR (2) = NM_UNSIGNED;
324 0451 NMLST_COUNT = 2;
325 0452 END;
326 0453
327 0454 END;
328 0455
329 0456 | At this point we've sorted out the following .NMLS dialects
330 0457 | .NMLS (no arguments)
331 0458 | .NMLS , (no arguments)
332 0459 | .NMLS foobar
333 0460 | .NMLS foobar
334 0461 | .NMLS 1, foobar
335 0462 | .NMLS 1 foobar
336 0463 | .NMLS 1,2
337 0464 | For all those cases, the value the next .LE command is to generate, if given,
338 0465 | is in NMLST_VALUE (2), and the list depth is in NMLST_VALUE (1).
339 0466
340 0467 | Check for just a single number, nothing following.
341 0468
342 0469 IF .NMLST_COUNT EQL 1
343 0470 THEN
344 0471 | Move the counter to the second position.
345 0472 BEGIN
346 0473 NMLST_VALUE (2) = .NMLST_VALUE (1);
347 0474 NMLST_DESCR (2) = .NMLST_DESCR (1);
348 0475 NMLST_DESCR (1) = NM_NULL; ! Cause the following code to ignore first argument.
```

NM
V04-000

Processes the various .NUMBER directives.
NM -- main routine

D 5
16-Sep-1984 01:16:58
14-Sep-1984 13:07:25

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]NM.BLI;1

Page 8
(4)

349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405

0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500
0501
0502
0503
0504
0505
0506
0507
0508
0509
0510
0511
0512
0513
0514
0515
0516
0517
0518
0519
0520
0521
0522
0523
0524
0525
0526
0527
0528
0529
0530
0531
0532

```
END;

! At this point all the valid variations of .NMLS have been sorted out
! and the arguments have been put in NMLST_VALUE (1) and NMLST_VALUE (2).
! From this point on, RUNOFF thinks the user said .NMLS n,m.

! Process first parameter, that indicates which
! list element counter is being set.
CASE .NMLST_DESCR (1) FROM 0 TO 4 OF
  SET
    [NM_BAD] :
      RETURN;

    [NM_PLUS, NM_MINUS] :
      NMLST_VALUE (1) = .NMLST_VALUE (1) + .LSTCNT [CL_INDEX];

    [NM_NULL] :
      NMLST_VALUE (1) = .LSTCNT [CL_INDEX];

    [NM_UNSIGNED] :
      0;

  TES;

  IF .NMLST_VALUE (1) LEQ 0 OR
    .NMLST_VALUE (1) GTR .LSTCNT [CL_MAX_INDEX]
  THEN
    BEGIN
      ERMA (RNFINM, FALSE);
      RETURN;
    END;

  LIST_DEPTH = .NMLST_VALUE (1);

  ! Process second parameter, that sets up the
  ! list element counter value.
  CASE .NMLST_DESCR (2) FROM 0 TO 4 OF
    SET
      [NM_BAD] :
        RETURN;

      [NM_NULL] :
        NMLST_VALUE (2) = .LSTCNT [.LIST_DEPTH] + 1;      ! Add 1 to offset the -1 later.

      [NM_PLUS, NM_MINUS] :
        NMLST_VALUE (2) = .NMLST_VALUE (2) + .LSTCNT [.LIST_DEPTH];

      [NM_UNSIGNED] :
        0;

    TES;

    IF .NMLST_VALUE (2) LSS 0
  THEN
    BEGIN
```



```

: 406      0533      4      ERMA (RNFINM, FALSE);
: 407      0534      4      RETURN;
: 408      0535      4      END;
: 409      0536      4
: 410      0537      4      ! When assigning the value, subtract one
: 411      0538      4      ! to anticipate the .LIST ELEMENT
: 412      0539      4      ! command that will increment it.
: 413      0540      4      LSTCNT [.LIST_DEPTH] = .NMLST_VALUE (2) - 1;
: 414      0541      4      END;
: 415      0542      4
: 416      0543      4      [H_NUMBER_SUBPAG, H_NUMBER_PAGE, H_NUMBER_RUNNIN] :
: 417      0544      4      BEGIN
: 418      0545      4
: 419      0546      4      LOCAL
: 420      0547      4      X;
: 421      0548      4      ! Copy of page/subpage/running-page to be updated
: 422      0549      4
: 423      0550      4      ! Copy the counter which is to be updated.
: 424      0551      4
: 425      0552      4      IF .PHAN_TOP_FIRST
: 426      0553      4      THEN
: 427      0554      4          ! Start with this page
: 428      0555      4          IF .HANDLER_CODE EQL H_NUMBER_SUBPAG
: 429      0556      4          THEN
: 430      0557      4              ! Copy subpage counter from current page
: 431      0558      4              X = .PAGEN [SCT_SUB_PAGE]
: 432      0559      4          ELSE
: 433      0560      4
: 434      0561      4              IF .HANDLER_CODE EQL H_NUMBER_RUNNIN
: 435      0562      4              THEN
: 436      0563      4                  ! Copy running page number for current page
: 437      0564      4                  X = .PAGEN [SCT_RUN_PAGE]
: 438      0565      4              ELSE
: 439      0566      4                  ! Copy page counter from current page
: 440      0567      4                  X = .PAGEN [SCT_PAGE]
: 441      0568      4          ELSE
: 442      0569      4
: 443      0570      4              ! Start it on next page
: 444      0571      4              IF .HANDLER_CODE EQL H_NUMBER_SUBPAG
: 445      0572      4              THEN
: 446      0573      4                  ! Get subpage counter from next page
: 447      0574      4                  X = .NPAGEN [SCT_SUB_PAGE]
: 448      0575      4              ELSE
: 449      0576      4
: 450      0577      4                  IF .HANDLER_CODE EQL H_NUMBER_RUNNIN
: 451      0578      4                  THEN
: 452      0579      4                      ! Get running page counter from next page
: 453      0580      4                      X = .NPAGEN [SCT_RUN_PAGE]
: 454      0581      4                  ELSE
: 455      0582      4                      ! Get page counter from next page
: 456      0583      4                      X = .NPAGEN [SCT_PAGE];
: 457      0584      4
: 458      0585      4      IF .NUM_RESULT
: 459      0586      4      THEN
: 460      0587      4      BEGIN
: 461      0588      4
: 462      0589      4      IF .NUM_SIGN NEQ 0
```

```

463 0590 4 THEN
464 0591 4 X = .X + .NUM_VALUE
465 0592 4 ELSE
466 0593 4
467 0594 4 IF .NUM_LENGTH NEQ 0
468 0595 4 THEN
469 0596 4 X = .NUM_VALUE
470 0597 4 ELSE
471 0598 4 ! No number was given. Try for letters.
472 0599 3 BEGIN
473 0600 3 FS_INIT (FS01); ! Initialize the temporary fixed string.
474 0601 3
475 0602 3 ! Now try to get a string of letters
476 0603 3 IF GSLU (IRA, FS01) EQL GSLU_NONE
477 0604 3 THEN
478 0605 3 ! It's okay to say .NUMBER PAGE and not give a page number.
479 0606 3 ! The net result is that all that happens is that page numbering
480 0607 3 ! is turned on again. (This happens as the result of the line of code
481 0608 3 ! just before the SELECT statement, above). However, for
482 0609 3 ! subpages, a number must be given, since 0 is the same as no
483 0610 3 ! subpage specified. In this case the program supplies a
484 0611 3 ! default value of 1.
485 0612 6 BEGIN
486 0613 6
487 0614 6 IF .HANDLER_CODE EQL H_NUMBER_SUBPAG
488 0615 6 THEN
489 0616 6 X = 1
490 0617 6 END
491 0618 3 ELSE
492 0619 3 ! Convert letters to the binary internal form.
493 0620 3 X = CONVLB (.FS_START (FS01), .FS_LENGTH (FS01))
494 0621 3 END
495 0622 3 END;
496 0623 3
497 0624 3 ! Validate the tentative page/subpage number
498 0625 3
499 0626 4 IF ((.HANDLER_CODE EQL H_NUMBER_SUBPAG) AND ! ZERO is illegal only for .NUMBER SUBPAGE
500 0627 3 (.X EQL 0)) OR
501 0628 4 (.X LSS 0) ! LSS 0 is always illegal
502 0629 3 THEN
503 0630 4 BEGIN
504 0631 4 ERMA (RNFIM, TRUE);
505 0632 4 RETURN
506 0633 3 END;
507 0634 3
508 0635 3 ! At this point X contains a valid counter.
509 0636 3 ! Now we need to put it where it will be picked up.
510 0637 3 IF .HANDLER_CODE EQL H_NUMBER_SUBPAG
511 0638 3 THEN
512 0639 4 BEGIN
513 0640 4
514 0641 4 ! Page numbers don't change while subpaging. However, check
515 0642 4 ! to see if the user has said .NUMBER PAGE in the interim.
516 0643 4 IF .NPAGEN [SCT_PAGE] - .PAGEN [SCT_PAGE] EQL 1
517 0644 4 THEN
518 0645 4 NPAGEN [SCT_PAGE] = .PAGEN [SCT_PAGE];
519 0646 4
```



```

520      0647 4      ! Save subpage counter
521      0648 4      IF .PHAN_TOP_FIRST
522      0649 4      THEN
523      0650 4          ! Subpage number takes effect immediately
524      0651 5          BEGIN
525      0652 5              PAGEN [SCT_SUB_PAGE] = .X;
526      0653 5              NPAGEN [SCT_SUB_PAGE] = .X + 1;
527      0654 5          END
528      0655 4      ELSE
529      0656 4          ! Subpage takes effect on next page
530      0657 4          NPAGEN [SCT_SUB_PAGE] = .X;
531      0658 4      END
532      0659 4      ELSE
533      0660 4          IF .HANDLER_CODE EQL H_NUMBER_RUNNIN
534      0661 4          THEN
535      0662 4              ! User said .NUMBER RUNNING
536      0663 4              IF .PHAN_TOP_FIRST
537      0664 4              THEN
538      0665 4                  ! The running page number takes effect immediately
539      0666 4                  BEGIN
540      0667 4                      PAGEN [SCT_RUN_PAGE] = .X;
541      0668 4                      NPAGEN [SCT_RUN_PAGE] = .X + 1;
542      0669 4                  END
543      0670 4              ELSE
544      0671 4                  ! The running page number takes effect on the next page
545      0672 4                  NPAGEN [SCT_RUN_PAGE] = .X
546      0673 4              END
547      0674 4          ELSE
548      0675 4              ! User said .NUMBER PAGE, not .NUMBER SUBPAGE or .NUMBER RUNNING
549      0676 4              BEGIN
550      0677 4                  HCT_ODD_EVEN = NOT (.X MOD 2);      ! Set odd/even page number flag appropriately.
551      0678 4              IF .PHAN_TOP_FIRST
552      0679 4              THEN
553      0680 4                  ! Page number takes effect immediately
554      0681 4                  BEGIN
555      0682 4                      PAGEN [SCT_PAGE] = .X;
556      0683 4                      NPAGEN [SCT_PAGE] = .X + 1;
557      0684 4                  END
558      0685 4              ELSE
559      0686 4                  ! Page number takes effect on the next page
560      0687 4                  NPAGEN [SCT_PAGE] = .X;
561      0688 4              END
562      0689 4          END;
563      0690 4      END;
564      0691 4      END;
565      0692 3      END;
566      0693 2      END;
567      0694 2      %IF DSRPLUS %THEN
568      0695 2          [H_NUMBER_EXAMPL, H_NUMBER_FIGURE, H_NUMBER_TABLE] :
569      0696 2          BEGIN
570      0697 2              LOCAL
571      0698 2              ENTITY_NUMBER,
572      0699 2              OFFSET;
573      0700 2          END
574      0701 2          ! Ignore command if an illegal number was given.
575      0702 2
576      0703 2
```

NM
V04-000

Processes the various .NUMBER directives.
NM -- main routine

H 5
16-Sep-1984 01:16:58
14-Sep-1984 13:07:25

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]NM.BLI;1

Page 12
(4)

**F

```

577 U 0704 2      IF NOT .NUM_RESULT
578 U 0705 2      THEN
579 U 0706 2      RETURN;
580 U 0707 2
581 U 0708 2      ! Assume user gives neither a number or a letter.
582 U 0709 2      ENTITY_NUMBER = 0;
583 U 0710 2
584 U 0711 2      ! Try to get a string of letters if the user didn't supply a number.
585 U 0712 2      IF .NUM_LENGTH EQL 0
586 U 0713 2      THEN
587 U 0714 2      BEGIN
588 U 0715 2      LOCAL
589 U 0716 2      GSLU_RESULT;
590 U 0717 2
591 U 0718 2      ! Initialize the fixed string where the result is returned.
592 U 0719 2      FS_INIT (FS01);
593 U 0720 2
594 U 0721 2      ! Now try to get a name specified as a string of letters.
595 U 0722 2      GSLU_RESULT = GSLU (IRA, FS01);
596 U 0723 2
597 U 0724 2      ! Convert to binary representation.
598 U 0725 2      IF .GSLU_RESULT NEQ GSLU_NONE
599 U 0726 2      THEN
600 U 0727 2      ENTITY_NUMBER = CONVLB (.FS_START (FS01), .FS_LENGTH (FS01));
601 U 0728 2
602 U 0729 2      END
603 U 0730 2      ELSE
604 U 0731 2      ENTITY_NUMBER = .NUM_VALUE;
605 U 0732 2
606 U 0733 2      ! Decide where to store the number.
607 U 0734 2      OFFSET = (SELECTONE .HANDLER_CODE OF
608 U 0735 2      SET
609 U 0736 2      [H_NUMBER_EXAMPL] : EXAMP_OFFSET;
610 U 0737 2      [H_NUMBER_FIGURE] : FIGUR_OFFSET;
611 U 0738 2      [H_NUMBER_TABLE] : TABLE_OFFSET;
612 U 0739 2      TES);
613 U 0740 2
614 U 0741 2      ! The "-1" in the statements below exists because HEADER.BLI
615 U 0742 2      ! increments before putting out the next entity caption.
616 U 0743 2
617 U 0744 2      IF .NUM_SIGN EQL 0
618 U 0745 2      THEN
619 U 0746 2      ! Unsigned number: just store it.
620 U 0747 2      !
621 U 0748 2      !
622 U 0749 2      ECC [.OFFSET, ECC$H_COUNTER] = .ENTITY_NUMBER - 1
623 U 0750 2      ELSE
624 U 0751 2      ! +n or -n: add it.
625 U 0752 2      !
626 U 0753 2      !
627 U 0754 2      ECC [.OFFSET, ECC$H_COUNTER] =
628 U 0755 2      .ECC [.OFFSET, ECC$H_COUNTER] + .ENTITY_NUMBER - 1;
629 U 0756 2
630 U 0757 2      IF .ECC [.OFFSET, ECC$H_COUNTER] LSS 0      ! Defensive check.
631 U 0758 2      THEN
632 U 0759 2      ECC [.OFFSET, ECC$H_COUNTER] = 0;
633 U 0760 2
```


NM
V04-000

Processes the various .NUMBER directives.
NM -- main routine

1 5
16-Sep-1984 01:16:58
14-Sep-1984 13:07:25

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]NM.BLI;1

Page 13
(4)

NOT

```
: 634      U 0761 2      END;
: 635      0762 2 %FI
: 636      0763 2
: 637      0764 2      TES;
: 638      0765 2
: 639      0766 1      END;
```

! End of NM

.TITLE NM Processes the various .NUMBER directives.
.IDENT \V04-000\

.EXTRN ECC, FS01, GCA, HCT
.EXTRN HLLIST, IRA, LSTCNT
.EXTRN NMLST, NPAGEN, NUMPRM
.EXTRN PAGEN, PHAN, RNFILC
.EXTRN RNFINM, GETLET, CONVLB
.EXTRN ERMA, GLNM, GSLU
.EXTRN RSKIPS

.PSECT \$CODE\$,NOWRT,2

		OFFC 00000	.ENTRY NM, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	: 0249
	5B 00000000G	EF 9E 00002	MOVAB LSTCNT, R11	
	5A 00000000G	EF 9E 00009	MOVAB HLLIST+4, R10	
	59 00000000G	EF 9E 00010	MOVAB PAGEN+8, R9	
	58 00000000G	EF 9E 00017	MOVAB NUMPRM+12, R8	
	57 00000000G	EF 9E 0001E	MOVAB NPAGEN+8, R7	
	56 00000000G	EF 9E 00025	MOVAB FS01, R6	
	55 00000000G	EF 9E 0002C	MOVAB NMLST, R5	
	53 04	AC D0 00033	MOVL HANDLER CODE, R3	: 0277
000000A2	8F	53 D1 00037	CMPL R3, #162	
		2E 13 0003E	BEQL 2\$	
000000A1	8F	53 D1 00040	CMPL R3, #161	: 0278
		25 13 00047	BEQL 2\$	
000000A4	8F	53 D1 00049	CMPL R3, #164	: 0279
		1C 13 00050	BEQL 2\$	
00000000G	FF	01 D0 00052	MOVL #1, @HCT+44	: 0282
	07 00000000G	EF E8 00059	BLBS PHAN+24, 1\$: 0287
	07 00000000G	EF E8 00060	BLBS PHAN, 2\$	
00000000G	FF	01 D0 00067 1\$:	MOVL #1, @HCT+12	: 0289
0000009B	8F	53 D1 0006E 2\$:	CMPL R3, #155	: 0296
		EF 19 00075	BLSS 10\$	
0000009C	8F	53 D1 00077	CMPL R3, #156	
		66 14 0007E	BGTR 10\$	
	01 F4	A8 E8 00080	BLBS NUMPRM, 3\$: 0303
		04 00084	RET	
		52 D4 00085 3\$:	CLRL SECTION_NUMBER	: 0307
		68 D5 00087	TSTL NUMPRM+T2	: 0310
		12 12 00089	BNEQ 4\$	
		58 DD 0008B	PUSHL R8	: 0320
	F8	A8 9F 0008D	PUSHAB NUMPRM+4	
	00000000G	EF 9F 00090	PUSHAB IRA	
00000000G	EF	03 FB 00096	CALLS #3, GETLET	
	52 F8	A8 D0 0009D 4\$:	MOVL NUMPRM+4, SECTION_NUMBER	: 0323
		05 12 000A1	BNEQ 5\$: 0325
		68 D5 000A3	TSTL NUMPRM+12	: 0326
		01 12 000A5	BNEQ 5\$	

				04	000A7		RET			
		1F	FC	A8	E9	000A8	5\$:	BLBC	NUMPRM+8, 8\$	0334
	0000009B	8F		53	D1	000AC		CMPL	R3, #155	0337
				0A	12	000B3		BNEQ	6\$	
		52	00000000GEF	42	9E	000B5		MOVAB	ECC+411[SECTION_NUMBER], SECTION_NUMBER	0338
				08	11	000BD		BRB	7\$	
		52	00000000GEF	42	9E	000BF	6\$:	MOVAB	ECC+375[SECTION_NUMBER], SECTION_NUMBER	0340
				02	18	000C7	7\$:	BGEQ	8\$	0341
				52	D4	000C9		CLRL	SECTION_NUMBER	
	0000009B	8F		53	D1	000CD	8\$:	DECL	R2	0347
				08	12	000D4		CMPL	R3, #155	0345
	00000000G	EF		52	D0	000D6		BNEQ	9\$	
				04	00	000DD		MOVL	R2, ECC+411	0347
	00000000G	EF		52	D0	000DE	9\$:	RET		0349
				04	00	000E5		MOVL	R2, ECC+375	0293
	000000A0	8F		53	D1	000E6	10\$:	RET		0353
				1A	12	000ED		CMPL	R3, #160	
		09	00000000G	EF	E9	000EF		BNEQ	12\$	
				A9	D4	000F6		BLBC	PHAN+24, 11\$	0359
F8	A9	04		00	F0	000F9		CLRL	PAGEN+4	0363
				02	F0	000FF	11\$:	INSV	#2, #0, #4, PAGEN	0364
F8	A7	04		00	F0	00102		CLRL	NPAGEN+4	0367
				04	00	00108		INSV	#2, #0, #4, NPAGEN	0368
	000000A1	8F		53	D1	00109	12\$:	RET		0293
				5B	12	00110		CMPL	R3, #161	0371
				AA	DD	00112		BNEQ	19\$	
	00000000G	EF		01	FB	00115		PUSHL	HLLIST	0374
		54		65	D0	0011C		CALLS	#1, GLNM	
				52	D4	0011F		MOVL	NMLST, R4	0376
				3F	11	00121		CLRL	I	
		6A		52	D0	00123	13\$:	BRB	18\$	
		00		CF	00	00126		MOVL	I, HLLIST+4	0378
0013	04	000C	00A0	C542	CF	00126	14\$:	CASEL	NMLST+160[I], #0, #4	0380
				0035		00120		.WORD	18\$-14\$,-	
				0035		00135			15\$-14\$,-	
									16\$-14\$,-	
									16\$-14\$,-	
									18\$-14\$	
				29	11	00137		BRB	18\$	
	6A42			6542	D0	00139	15\$:	MOVL	NMLST[I], HLLIST+4[I]	0387
				22	11	0013E		BRB	18\$	
	51			6542	D0	00140	16\$:	MOVL	NMLST[I], R1	0394
	50			6142	DE	00144		MOVAL	(R1)[I], R0	
	50			5A	C0	00148		ADDL2	R10, R0	
				06	19	0014B		BLSS	17\$	
	6A42			51	C0	0014D		ADDL2	R1, HLLIST+4[I]	0396
				0F	11	00151		BRB	18\$	
				7E	D4	00153	17\$:	CLRL	-(SP)	0398
				8F	DD	00155		PUSHL	#RNF INM	
	00000000G	EF		02	FB	0015B		CALLS	#2, ERMA	
BD		52		54	F3	00162	18\$:	AOBLEQ	R4, I, 13\$	0376
		50		6A	D0	00166		MOVL	HLLIST+4, R0	0404
				6A40	D7	00169		DECL	HLLIST+4[R0]	
				04	00	0016C		RET		0293
	000000A2	8F		53	D1	0016D	19\$:	CMPL	R3, #162	0407
				03	13	00174		BEQL	20\$	
				00DF	31	00176		BRW	34\$	

NM
V04-000

Processes the various .NUMBER directives.
NM -- main routine

K 5
16-Sep-1984 01:16:58
14-Sep-1984 13:07:25

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]NM.BLI;1

Page 15
(4)

		00A4	C5	7C	00179	20\$:	CLRQ	NMLST+164	0413		
			02	DD	0017D		PUSHL	#2	0415		
	00000000G	EF	01	FB	0017F		CALLS	#1, GLNM			
		50	65	DD	00186		MOVL	NMLST, R0	0431		
			10	13	00189		BEQL	21\$			
		01	50	D1	0018B		CMPL	R0, #1	0432		
			0B	13	0018E		BEQL	21\$			
		02	50	D1	00190		CMPL	R0, #2	0433		
			3D	12	00193		BNEQ	22\$			
			00A8	C5	D5	00195	TSTL	NMLST+168	0434		
			37	12	00199		BNEQ	22\$			
			0C	A6	D4	0019B	21\$:	CLRL	FS01+12	0439	
		66	10	A6	9E	0019E	MOVAB	FS01+16, FS01			
	04	A6	66	DD	001A2		MOVL	FS01, FS01+4			
			56	DD	001A6		PUSHL	R6	0443		
	00000000G	EF	9F	001A8			PUSHAB	IRA			
		01	02	FB	001AE		CALLS	#2, GSLU			
			50	D1	001B5		CMPL	R0, #1			
			18	12	001B8		BNEQ	22\$			
			0C	A6	DD	001BA	PUSHL	FS01+12	0448		
			66	DD	001BD		PUSHL	FS01			
	00000000G	EF	02	FB	001BF		CALLS	#2, CONVLB			
		08	A5	50	DD	001C6	MOVL	R0, NMLST+8			
		00A8	C5	01	DD	001CA	MOVL	#1, NMLST+168	0450		
			65	02	DD	001CF	MOVL	#2, NMLST	0451		
		01	65	D1	001D2	22\$:	CMPL	NMLST, #1	0469		
			10	12	001D5		BNEQ	23\$			
		08	A5	04	A5	DD	001D7	MOVL	NMLST+4, NMLST+8	0473	
		00A8	C5	00A4	C5	DD	001DC	MOVL	NMLST+164, NMLST+168	0474	
				00A4	C5	D4	001E3	CLRL	NMLST+164	0475	
				00A4	C5	CF	001E7	23\$:	CASEL	NMLST+164, #0, #4	0484
000B		04	00		0015	001ED	24\$:	.WORD	26\$-24\$,-		
	000B	001D	019C	001F5					27\$-24\$,-		
									25\$-24\$,-		
									25\$-24\$,-		
									57\$-24\$		
				04	001F7		RET		0488		
		50	6B	DD	001F8	25\$:	MOVL	LSTCNT, R0	0491		
	04	A5	04	A0	CO	001FB	ADDL2	4(R0), NMLST+4			
				08	11	00200	BRB	27\$			
		50	6B	DD	00202	26\$:	MOVL	LSTCNT, R0	0494		
	04	A5	04	A0	DD	00205	MOVL	4(R0), NMLST+4			
		50	04	A5	DD	0020A	27\$:	MOVL	NMLST+4, R0	0501	
				37	15	0020E	BLEQ	32\$			
	00	BB		50	D1	00210	CMPL	R0, @LSTCNT	0502		
				31	14	00214	BGTR	32\$			
		52	04	A5	DD	00216	MOVL	NMLST+4, LIST_DEPTH	0509		
		00	00A8	C5	CF	0021A	CASEL	NMLST+168, #0, #4	0513		
0018		04	000E	00220	28\$:	.WORD		29\$-28\$,-			
	0018	0022	0169	00228				31\$-28\$,-			
								30\$-28\$,-			
								30\$-28\$,-			
								57\$-28\$			
				04	0022A		RET		0517		
		50	00	BB42	DE	0022B	29\$:	MOVAL	@LSTCNT[LIST_DEPTH], R0	0520	
	08	A5	04	A0	01	C1	00230	ADDL3	#1, 4(R0), NMLST+8		
				0A	11	00236	BRB	31\$			

NOT
V04

NM
V04-000

Processes the various .NUMBER directives.
NM -- main routine

L 5
16-Sep-1984 01:16:58
14-Sep-1984 13:07:25

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]NM.BLI;1

Page 16
(4)

	08	50	00	BB42	DE	00238	30%:	MOVAL	@LSTCNT[LIST_DEPTH], R0		0523
		A5	04	A0	C0	0023D		ADDL2	4(R0), NMLST+8		
			08	A5	D5	00242	31%:	TSTL	NMLST+8		0530
				05	18	00245		BGEQ	33\$		
				7E	D4	00247	32%:	CLRL	-(SP)		0533
				00CD	31	00249		BRW	49\$		
04	A0	08	50	00	BB42	DE	0024C	33%:	MOVAL	@LSTCNT[LIST_DEPTH], R0	0540
			A5	01	C3	00251		SUBL3	#1, NMLST+8, -4(R0)		
					04	00257		RET			0293
	000000A3	8F		53	D1	00258	34%:	CMPL	R3, #163		0543
				01	18	0025F		BGEQ	35\$		
					04	00261		RET			
	000000A5	8F		53	D1	00262	35%:	CMPL	R3, #165		
				01	15	00269		BLEQ	36\$		
					04	0026B		RET			
	27	00000000G		EF	E9	0026C	36%:	BLBC	PHAN+24, 39\$		0551
				54	D4	00273		CLRL	R4		0555
	000000A5	8F		53	D1	00275		CMPL	R3, #165		
				08	12	0027C		BNEQ	37\$		
				54	D6	0027E		INCL	R4		
	52		FA	A9	3C	00280		MOVZWL	PAGEN+2, X		0558
				39	11	00284		BRB	42\$		
	000000A4	8F		53	D1	00286	37%:	CMPL	R3, #164		0561
				06	12	0028D		BNEQ	38\$		
	52		06	A9	3C	0028F		MOVZWL	PAGEN+14, X		0564
				2A	11	00293		BRB	42\$		
	52			69	D0	00295	38%:	MOVL	PAGEN+8, X		0567
				25	11	00298		BRB	42\$		0555
				54	D4	0029A	39%:	CLRL	R4		0571
	000000A5	8F		53	D1	0029C		CMPL	R3, #165		
				08	12	002A3		BNEQ	40\$		
				54	D6	002A5		INCL	R4		
	52		FA	A7	3C	002A7		MOVZWL	NPAGEN+2, X		0574
				12	11	002AB		BRB	42\$		
	000000A4	8F		53	D1	002AD	40%:	CMPL	R3, #164		0577
				06	12	002B4		BNEQ	41\$		
	52		06	A7	3C	002B6		MOVZWL	NPAGEN+14, X		0580
				03	11	002BA		BRB	42\$		
	52			67	D0	002BC	41%:	MOVL	NPAGEN+8, X		0583
	4B		F4	A8	E9	002BF	42%:	BLBC	NUMPRM, 46\$		0585
			FC	A8	D5	002C3		TSTL	NUMPRM+8		0589
				06	13	002C6		BEQL	43\$		
	52		F8	A8	C0	002C8		ADDL2	NUMPRM+4, X		0591
				40	11	002CC		BRB	46\$		
				68	D5	002CE	43%:	TSTL	NUMPRM+12		0594
				06	13	002D0		BEQL	44\$		
	52		F8	A8	D0	002D2		MOVL	NUMPRM+4, X		0596
				36	11	002D6		BRB	46\$		
			0C	A6	D4	002D8	44%:	CLRL	FS01+12		0600
			10	A6	9E	002DB		MOVAB	FS01+16, FS01		
04	66			66	D0	002DF		MOVL	FS01, FS01+4		
	A6			56	DD	002E3		PUSHL	R6		0603
				EF	9F	002E5		PUSHAB	IRA		
	00000000G	EF		02	FB	002EB		CALLS	#2, GSLU		
		02		50	D1	002F2		CMPL	R0, #2		
				08	12	002F5		BNEQ	45\$		
				19	54	E9	002F7	BLBC	R4, 47\$		0614

NOT
V04

NM
V04-000

Processes the various .NUMBER directives.
NM -- main routine

M 5
16-Sep-1984 01:16:58
14-Sep-1984 13:07:25

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]NM.BLI;1

Page 17
(4)

NOT
V04

	52		01	DO	002FA	MOVL	#1, X	:	0616			
			0F	11	002FD	BRB	46\$:	0612			
		OC	A6	DD	002FF	45\$:	PUSHL	FS01+12	:	0620		
			66	DD	00302		PUSHL	FS01	:			
	00000000G	EF	02	FB	00304		CALLS	#2, CONVLB	:			
		52	50	DO	0030B		MOVL	R0, X	:			
		02	54	E9	0030E	46\$:	BLBC	R4, 47\$:	0626		
			04	13	00311		BEQL	48\$:	0627		
			52	D5	00313	47\$:	TSTL	X	:	0628		
			10	18	00315		BGEQ	50\$:			
			01	DD	00317	48\$:	PUSHL	#1	:	0631		
			8F	DD	00319	49\$:	PUSHL	#RNF INM	:			
	00000000G	EF	02	FB	0031F		CALLS	#2, ERMA	:			
				04	00326		RET		:	0630		
		51	00000000G	EF	DO	00327	50\$:	MOVL	PHAN+24, R1	:	0648	
		1E		54	E9	0032E		BLBC	R4, 53\$:	0652	
	50	69		01	C1	00331		ADDL3	#1, PAGEN+8, R0	:	0643	
		50		67	D1	00335		CMPL	NPAGEN+8, R0	:		
				03	12	00338		BNEQ	51\$:		
		67		69	DO	0033A		MOVL	PAGEN+8, NPAGEN+8	:	0645	
		0A		51	E9	0033D	51\$:	BLBC	R1, 52\$:	0648	
	FA	A7	FA	52	BO	00340		MOVW	X, PAGEN+2	:	0652	
				01	A1	00344		ADDW3	#1, X, NPAGEN+2	:	0653	
					04	00349		RET		:	0648	
		FA	A7	52	BO	0034A	52\$:	MOVW	X, NPAGEN+2	:	0657	
					04	0034E		RET		:	0637	
		000000A4	8F	53	D1	0034F	53\$:	CMPL	R3, #164	:	0661	
				12	12	00356		BNEQ	55\$:		
		0A		51	E9	00358		BLBC	R1, 54\$:	0665	
		06		52	BO	0035B		MOVW	X, PAGEN+14	:	0669	
	06	A7		01	A1	0035F		ADDW3	#1, X, NPAGEN+14	:	0670	
					04	00364		RET		:	0665	
		06	A7	52	BO	00365	54\$:	MOVW	X, NPAGEN+14	:	0674	
					04	00369		RET		:	0665	
7E				01	7A	0036A	55\$:	EMUL	#1, X, #0, -(SP)	:	0679	
50	00			02	7B	0036F		EDIV	#2, (SP)+, R0, R0	:		
	50			50	D2	00374		MCOML	R0, HCT+16	:		
		00000000G	EF	51	E9	0037B		BLBC	R1, 56\$:	0681	
			08	52	DO	0037E		MOVL	X, PAGEN+8	:	0685	
			69		9E	00381		MOVAB	1(R2), NPAGEN+8	:	0686	
			67	01	04	00385		RET		:	0681	
					52	DO	00386	56\$:	MOVL	X, NPAGEN+8	:	0690
			67		04	00389	57\$:	RET		:	0766	

; Routine Size: 906 bytes, Routine Base: \$CODE\$ + 0000

; 640 0767 1
; 641 0768 1 END
; 642 0769 0 ELUDOM

! End of module

PSECT SUMMARY

NM
V04-000

Processes the various .NUMBER directives.
NM -- main routine

N 5
16-Sep-1984 01:16:58
14-Sep-1984 13:07:25

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]NM.BLI;1

Page 18
(4)

```

:
: Name                      Bytes                      Attributes
:
: $CODE$                   906 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)
:

```

Library Statistics

```

:
: File                      Total      Symbols  Percent      Pages      Processing
:                               Loaded      Time
:
: $255$DUA28:[SYSLIB]XPORT.L32;1      590          0          0        252        00:00.1
: _$255$DUA28:[RUNOFF.SRC]DSRLIB.L32;1 1248         86          6         86        00:00.2
:

```

COMMAND QUALIFIERS

```

:
: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:NM/OBJ=OBJ$:NM MSRC$:NM/UPDATE=(ENH$:NM)
:
: Size:          906 code + 0 data bytes
: Run Time:      00:18.6
: Elapsed Time:  00:38.4
: Lines/CPU Min: 2483
: Lexemes/CPU-Min: 21435
: Memory Used:   212 pages
: Compilation Complete

```


0346 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

